(1)

50272-101

REPORT DOCUMENTATION PAGE	1. REPORT NO. DCA/SW/MT-88/001n	2.	3. Recipient's Accession No.

4. Title and Subtitle

Defense Communications Agency
Upper Level Protocol Test System
Transmission Control Protocol / Internet Protocol (Tightly Coupled)

5. Report Date
May 1988

6.

7. Author(s) Test Traceability Index

8. Performing Organization Rept. No.

9. Performing Organization Name and Address

Defense Communications Agency
Defense Communications Engineering Center
Code R640
1860 Wiehle Ave.
Reston, VA 22090-5500

10. Project/Task/Work Unit No.

11. Contract(C) or Grant(G) No.
(C)
(G)

12. Sponsoring Organization Name and Address

13. Type of Report & Period Covered
FINAL

14.

15. Supplementary Notes

For magnetic tape, see: AD-A195 128.

16. Abstract (Limit: 200 words)

This document is part of a software package that provides the capability to conformance test the Department of Defense suite of upper level protocols including: Internet Protocol (IP) Mil-Std 1777, Transmission Cont·ol Protocol (TCP) Mil-Std 1778, File Transfer Protocol (FTP) Mil-Std 1780, Simple Mail Transfer Protocol (SMTP) Mil-Std 1781 and TELNET Protocol Mil-Std 1782. See p 1

17. Document Analysis a. Descriptors

Protocol Test Systems
Conformance Testing
Department of Defense Protocol Suite

b. Identifiers/Open-Ended Terms

Internet Protocol (IP)
Transmission Control Protocol (TCP)
File Transfer Protocol (FTP)
Simple Mail Transfer Protocol (SMTP)
TELNET Protocol

c. COSATI Field/Group

18. Availability Statement Unlimited Release	19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages 70
	20. Security Class (This Page) UNCLASSIFIED	22. Price

88 7 06 097

DEFENSE COMMUNICATIONS AGENCY

UPPER LEVEL PROTOCOL TEST SYSTEM

TRANSMISSION CONTROL PROTOCOL/INTERNET PROTOCOL
MIL-STD 1778 AND MIL-STD 1777
(TIGHTLY COUPLED)
TEST TRACEABILITY INDEX

MAY 1988

Disclaimer Concerning Warranty and Liability

This software product and documentation and all future updates to it are provided by the United States Government and the Defense Communications Agency (DCA) for the intended purpose of conducting conformance tests for the DoD suite of higher level protocols. DCA has performed a review and analysis of the product along with tests aimed at insuring the quality of the product, but does not warranty or make any claim as to the quality of this product. The product is provided "as is" without warranty of any kind, either expressed or implied. The user and any potential third parties accept the entire risk for the use, selection, quality, results, and performance of the product and updates. Should the product or updates prove to be defective, inadequate to perform the required tasks, or misrepresented, the resultant damage and any liability or expenses incurred as a result thereof must be borne by the user and/or any third parties involved, but not by the United States Government, including the Department of Commerce and/or The Defense Communications Agency and/or any of their employees or contractors.

Distribution and Copyright

Comments

Comments or questions about this software product and documentation can be addressed in writing to:

DCA Code R640
1860 Wiehle Ave
Reston, VA 22090-5500
ATTN: Protocol Test System Administrator

TRANSMISSION CONTROL PROTOCOL/INTERNET PROTOCOL (TCP/IP)
MIL-STD-1778 and MIL-STD-1777
TRACEABILITY MATRIX

This Traceability Matrix provides information on the derivation, organization, and function of tests specified for TCP/IP within the Protocol Test System.

The document is divided into four sections:

TCP/IP TRACEABILITY INDEX;
TCP/IP TEST INDEX;
TCP/IP TEST SCENARIOS INDEX;
TCP/IP SCENARIOS AND TEST DESCRIPTIONS.

TCP/IP TRACEABILITY INDEX: TCP/IP TEST NUMBERS VERSUS TCP MIL-STD-1778 AND IP MIL-STD-1777 REFERENCES . . .

The table indicates the cross-reference between the Test Scenarios and the applicable sections in MIL-STD-1778 and MIL-STD-1777 regarding each required function, operation, option, mode, response, or state.

TCP/IP TEST INDEX: TCP/IP TEST NUMBERS VERSUS TCP COMMANDS/PRIMITIVES/OPTIONS/MODES AND IP COMMANDS/PRIMITIVES/OPTIONS/MODES . . .

The table shows the TCP/IP Test Numbers that may be regarded as the "principle tests" of: each TCP Command or Primitive and Option or Mode; each IP Command or Primitive and Option or Mode.

TCP/IP TEST SCENARIOS INDEX: TCP/IP TEST SCENARIO FILES VERSUS TCP/IP TEST NUMBERS . . .

The table shows, for each TCP/IP Test Number, the UNIX file name of the TCP/IP Test Scenario File in which that number appears.

TCP/IP SCENARIOS AND TEST DESCRIPTIONS . . .

This section provides a brief narrative of the scope and objectives of each TCP/IP Test Scenario File and an operational description of each TCP/IP Test Number.

SECTION 1 - TCP/IP TRACEABILITY INDEX

TCP/IP Test Numbers Versus IP MIL-STD-1777 and TCP MIL-STD-1778

The table indicates the cross-reference between the TCP/IP tests and the applicable sections of MIL-STD-1777 and MIL-STD-1778.

Reference		**Test Number**
MIL-STD-1777:		
6.2.1.1	Send Service Request	101
6.2.2.1	Deliver Service Response	101
9.3.3	Type of Service Parameters	102, 103-106
9.3.3 9.4.6.3.2	Precedence Values	102, 136
9.3.3	Low Delay - Acceptance	103
9.3.3	High Reliability - Acceptance	104
9.3.3	High Throughput - Acceptance	106
9.4.6.2.8 9.3.8	Time to Live	107, 108 109
9.4.6.2.7	Invalid Version Number	110
9.2.3	Checksum	111
9.3.2 9.4.6.2.7	Header Length	112, 113
9.3.2 9.3.4	Total Length	113-116
9.4.6.3.10	More Fragments Field	117
9.3.15.1 9.3.15.2	NOP and End-of-Options List Options	118-121
9.3.13.1	Datagram with Invalid Options - Rejection	122
9.4.6.3.10	Reassembly of Valid Datagrams	123-127, 129, 132-135
9.4.6.3.11	Time to Live in Reassembly	128, 132, 133
9.4.6.3.9	Inconsistent Fragment Parameters	130, 131
9.4.6.3.1	Echo, Timestamp, and Information ICMP Messages	137-139

Reference		Test Number
MIL-STD 1778:		
TCP Service Request Primitives		
6.4.1	Unspecified Passive Open	1, 39, 57
6.4.2	Fully Specified Passive Open	9, 10, 40
6.4.3	Active Open	2, 38, 55
6.4.4	Active Open with Data	11, 12
6.4.5	Send	3, 14-16, 56
6.4.6	Allocate	51
6.4.7	Close	6, 7, 14-16
6.4.8	Abort	17-20
6.4.9	Status	24
TCP Service Response Primitives		
6.4.10.1	Open ID	1, 2, 9-11
6.4.10.2	Open Failure	35, 371, 41, 42, 49, 63
6.4.10.3	Open Success	1, 2, 66
6.4.10.4	Deliver	3, 6
6.4.10.5	Closing	6, 7
6.4.10.6	Terminate	6, 7, 17-21, 44-46, 55-58, 67
6.4.10.7	Status Response	24
6.4.10.8	Error	4, 5, 31, 38, 40, 73
TCP Mechanisms Tested		
9.2.3	Flow Control Window	59, 61
9.2.4	Duplicate/Order Detection	25-27, 29, 32
9.2.5	ACKs and Retransmission	27, 52-54
9.2.6	Checksum	28
9.2.7	Push	63, 64
9.2.8	Urgent	60-62
9.2.9	ULP Timeout/Action	55-58
9.2.10	Security	38-50
9.2.11	Precedence Level	21-23
9.2.12	Multiplexing	30-34, 36
9.2.13	Connection Opening	35, 37
9.2.14	Connection Closing	14-17
9.2.15	Resets	35, 50, 65-72
9.3.1	Source Port Address Range	13
9.3.2	Destination Port Address Range	13
TCP Options Tested		
9.3.11.1.3	Maximum Segment Size	52

SECTION 2 - TCP/IP TEST INDEX

The table shows the TCP/IP Test Numbers that may be regarded as the "principle tests" for each TCP/IP service request, response, and option.

Test Number	**Purpose**
IP Tests:	
101	Deliver and Send Datagram
102	Precedence Values - Acceptance
103	Low Delay - Acceptance104
104	High Reliability - Acceptance
105	High Throughput - Acceptance
106	Type of Service Combinations - Acceptance
107	Illegal Time to Live - Rejection
108	Too Small Time to Live - Rejection
109	Range of Valid Time to Live Values - Acceptance
110	Invalid Version Number - Rejection
111	Invalid Checksum - Rejection
112	Illegally Small Header Length - Rejection
113	Inconsistent Header and Total Length - Rejection
114	Illegally Small Total Length - Rejection
115	Total Length Greater Than Actual Length - Rejection
116	Total Length Smaller Than Actual Length - Rejection
117	More Fragments Field - Recognition
118	Datagram with NOP and End-of-Options List Options - Acceptance
119	Datagram with 2 NOP, 1 End-of-Options List Options - Acceptance
120	Datagram with 3 NOP, 1 End-of-Options List Options - Acceptance
121	Datagram with 4 NOP Options - Acceptance
118	Datagram with Invalid Options - Rejection
123	Reassembly of 2-Fragment Datagram
124	Reassembly of 3-Fragment Datagram
125	Reassembly of 576-Octet Datagram
126	Reassembly of Out-of-Order Fragments - Mixed
127	Reassembly of Fragments Received in Reverse Order
128	Expired Time to Live in Arriving Fragment - Rejection

Test Number	Purpose
129	Duplicate Fragment in Reassembly
130	Inconsistent Protocol Fields in Fragment Reassembly - Rejection
131	Inconsistent Precedence Fields in Fragment
132	Expiration of Time to Live during Reassembly - Rejection Reassembly - Rejection
133	Setting and Restarting Reassembly Timer
134	Reassembly of Two Intermixed Datagrams
135	Reassembly of Many Intermixed Datagrams
136	Precedence Values - Transmission
137	Echo and Echo Reply
138	Timestamp and Timestamp Reply
139	Information Request and Information Reply
TCP Tests:	
1	Unspecified Passive Open Request
2	Active Open Request
3	Basic Data Transfer
4	Remote Driver Interpretation of Command LCN
5	Determine IUT Standard Send Buffer
6	Closing Handshake - IUT initiates close
7	Closing Handshake - IUT peer initiates close
8	Ability to Reconnect Remote Driver Command Channel
9	Fully Specified Passive Open Request
10	Illegal Fully Specified Passive Open Request
11	Active Open with Data
12	Active Open with Data
13	Port Number Range
14	Graceful Closing - Completion of data transfer after ULP close
15	Graceful Closing - Data transfer after receipt of peer's FIN
16	Graceful Closing - Peer data transfer after IUT initiates close
17	ULP Abort
18	Peer Abort
19	ULP Abort - Data queued for sending
20	Peer Abort - Data queued for sending

Test Number	Purpose
21	Precedence - Mismatched
22	Precedence - Matched
23	Precedence Negotiation
24	Status
25	Out-of-Order Data
26	Overlapping Data
27	Lost Data
28	TCP Bad Checksum Detection
29	Sequence Number Wraparound
30	Multiplexing - Two connections with unique 4-tuple IUT opens passively
31	Multiplexing - Common destination port in 4-tuple common IUT port
32	Multiplexing - Common destination port in 4-tuple common REF port
33	Multiplexing - Two connections with unique 4-tuple IUT opens actively
34	Multiplexing - Three connections with common IUT port in 4-tuple
35	Duplicate Connection Attempt - IUT Passive
36	Multiplexing - Same sequence numbers on two connections
37	Duplicate Connection Attempt - IUT Active
38	Setting Security in Active Open
39	Setting Security in Passive Open
40	Setting Security in Fully Specified Passive Open
41	Secure IUT rejecting connection to unsecured peer
42	Secure IUT rejecting connection from unsecured peer
43	Security option placement in sending data
44	Response to data with mismatched security class
45	Response to data with mismatched security protection authority
46	Response to data with extra protection authority
47	Use of security option for unclassified connections
48	Recognition of UNCLASS and GENSER as unsecured
49	Unsecured IUT response to connection attempt by secured host
50	Unsecured IUT response to data marked with classified security
51	Alloc

Test Number	Purpose
52	Maximum segment size option
53	Retransmission after acknowledgment of data
54	Retransmission after acknowledgment of SYN and FIN
55	ULP timeout service in Active Open
56	ULP timeout service in Send
57	ULP timeout service in Passive Open
58	ULP timeout notify action tested
59	TCP in window mechanism
60	Urgent service
61	Urgent service when peer has zero window
62	Urgent data delivery
63	Push service - Service not requested
64	Push service - Service requested
65	Reset - as response to connection refusal
66	Reset - partial reset prior to connection establishment
67	Reset - response to reset received while sending data
68	Reset segment format on receipt of Active Open with no listening port
69	Reset segment format on receipt of Active Open with data with no listening port
70	Reset segment format on receipt of invalid segment with ACK set
71	Reset segment format on receipt of invalid segment with SYN and ACK set
72	Reset - no reset sent on receipt of segment with bad acknowledgment number
73	Determine number of connections resources will allow

SECTION 3 - TCP/IP TEST SCENARIOS INDEX

The table shows, for each TCP/IP Test Number, the UNIX file name of the TCP/IP Scenario File in which it appears.

Test Number	**Scenario Name**
101	IP_BASIC
102	IP_BASIC
103	IP_BASIC
104	IP_BASIC
105	IP_BASIC
106	IP_BASIC
107	IP_BASIC
108	IP_BASIC
109	IP_BASIC
110	IP_BASIC
111	IP_BASIC
112	IP_BASIC
113	IP_BASIC
114	IP_BASIC
115	IP_BASIC
116	IP_BASIC
117	IP_BASIC
118	IP_BASIC
119	IP_BASIC
120	IP_BASIC
121	IP_BASIC
122	IP_BASIC
123	FRAGMENTS
124	FRAGMENTS
125	FRAGMENTS
126	FRAGMENTS
127	FRAGMENTS
128	FRAGMENTS
129	FRAGMENTS
130	FRAGMENTS
131	FRAGMENTS
132	EXTENDIP
133	EXTENDIP
134	EXTENDIP
135	EXTENDIP
136	EXTENDIP
137	ICMP
138	ICMP
139	ICMP

Test Number	Scenario Name
1	BASIC
2	BASIC
3	BASIC
4	BASIC
5	BASIC
6	BASIC
7	BASIC
8	BASIC
9	OPEN
10	OPEN
11	OPEN
12	OPEN
13	OPEN
22	OPEN
23	OPEN
24	OPEN
14	CLOSE
15	CLOSE
16	CLOSE
17	CLOSE
18	CLOSE
19	CLOSE
20	CLOSE
21	CLOSE
25	RELIABILITY
26	RELIABILITY
27	RELIABILITY
28	RELIABILITY
29	RELIABILITY
30	MULTIPLEX
31	MULTIPLEX
31	MULTIPLEX
32	MULTIPLEX
33	MULTIPLEX
34	MULTIPLEX
35	MULTIPLEX
36	MULTIPLEX
37	MULTIPLEX

Test Number	Scenario Name
38	SECURITY
39	SECURITY
40	SECURITY
41	SECURITY
42	SECURITY
43	SECURITY
44	SECURITY
45	SECURITY
46	SECURITY
47	SECURITY
48	SECURITY
49	SECURITY
50	SECURITY
51	ALLOC
52	POLICY
53	POLICY
54	POLICY
55	POLICY
56	POLICY
57	POLICY
58	POLICY
59	OUT_OF_BAND
60	OUT_OF_BAND
61	OUT_OF_BAND
62	OUT_OF_BAND
63	OUT_OF_BAND
64	OUT_OF_BAND
65	RESET
66	RESET
67	RESET
68	RESET
69	RESET
70	RESET
71	RESET
72	RESET
73	QUAL

SECTION 4 - TCP/IP SCENARIOS AND TEST DESCRIPTIONS

This section provides a brief narrative of the scope and objectives of each tightly coupled TCP/IP Test Scenario file and describes individual tests in each scenario. Tests numbered 100 and above test IP functions; tests numbered below 100 test TCP functions.

4A - IP SCENARIOS AND TEST DESCRIPTIONS

Scenario IP_BASIC

This scenario tests whether the Implementation Under Test (IUT) accepts all datagrams with valid header values and drops datagrams with invalid or inconsistent values.

TEST 101: CORRECT DEFAULT DATAGRAM

The IUT should accept a datagram formatted with all header fields set to correct default values.

- Action: The Central Driver (CD) will send a datagram correctly set with default values to the IUT. The datagram's data will be an Active Open to a non-listening port on the IUT.

- Verification: The IUT should return a reset.

- Success: The IUT returns a reset.

- Failure: The IUT does not return a reset.

TEST 102: ACCEPTANCE OF ALL PRECEDENCE VALUES

Determine that the IUT will accept datagrams with all precedence values.

- Action: CD will send a series of datagrams to the IUT. The datagram's data will be Active Opens to a non-listening port on the IUT. Precedence in the datagram will vary from 0 to 7. All other values in the header will be the default values.

- Verification: The IUT should return a reset for each datagram.

- Success: The IUT returns a reset for every datagram.

- Failure: The IUT does not return a reset for every datagram.

TEST 103: ACCEPTANCE OF LOW DELAY

Determine that the IUT will accept a datagram sent with low delay.

- Action: CD will send a datagram sent with low delay to the IUT. The datagram's data will be an Active Open to a non-listening port on the IUT. All other values in the header will be default values.

- Verification: The IUT should return a reset.

- Success: The IUT returns a reset.

- Failure: The IUT does not return a reset.

TEST 104: ACCEPTANCE OF HIGH RELIABILITY

Determine that the IUT will accept a datagram sent with high reliability.

- Action: CD will send a datagram set with high reliability to the IUT. The datagram's data will be an Active Open to a non-listening port on the IUT. All other values in the header will be default values.

- Verification: The IUT should return a reset.

- Success: The IUT returns a reset.

- Failure: The IUT does not return a reset.

TEST 105: ACCEPTANCE OF HIGH THROUGHPUT

Determine that the IUT accepts a datagram set with high throughput.

- Action: Using the TCP interface, CD will send the IUT a datagram set with high throughput. The datagram's data will be an Active Open to a non-listening port on the IUT. All other values in the header will be default values.

- Verification: The IUT should return a reset.

- Success: The IUT returns a reset.

- Failure: The IUT does not return a reset.

TEST 106: ACCEPTANCE OF TYPE OF SERVICE COMBINATIONS

Determine that the IUT accepts datagrams set with every combination of type of service -- precedence, delay, throughput, and reliability.

- Action: Using the TCP interface, CD will send the IUT a series of datagrams set with every combination of precedence, delay, throughput, and reliability. The IUT's data will be an Active Open to a non-listening port on the IUT. All other values in the IP header will be default values.

- Verification: The IUT should return a reset for every datagram sent.

- Success: The IUT returns a reset for every datagram sent.

- Failure: The IUT does not return a reset for every datagram sent.

TEST 107: RECOGNITION OF ILLEGALLY SMALL TIME TO LIVE

Determine that the IUT drops a datagram with an illegally small time to live (time to live of 0).

- Action: CD will send the IUT a datagram with a time to live of 0. The datagram's data will be an Active Open to a non-listening port on the IUT. All other datagram values will be valid default values.

- Verification: No response should be received from the IUT and the IUT accepts a subsequent valid datagram sent by the CD.

- Success: No response is received from the IUT and the IUT accepts the subsequent valid datagram sent by the CD.

- Failure: The IUT returns a reset.

TEST 108: RECOGNITION OF TOO SMALL TIME TO LIVE

Determine that the IUT drops a datagram arriving with time to live of 1.

- Action: CD will send the IUT a datagram with a time to live of 1. The datagram's data will be an Active Open to a non-listening port on the IUT. All other datagram values will be valid default values.

- Verification: No response should be received from the IUT and the IUT accepts a subsequent valid datagram sent by the CD.

- Success: No response is received from the IUT and the IUT accepts the subsequent valid datagram sent by the CD.

- Failure: The IUT returns a reset.

TEST 109: ACCEPTANCE OF RANGE OF TIME TO LIVE VALUES

Determine that the IUT accepts datagrams set with the upper and lower bounds of valid time to live values.

- Action: CD will send the IUT a datagram with a time to live of 2. The datagram's data will be an Active Open to a non-listening port on the IUT. All other datagram values will be valid default values. The CD will send the IUT a second datagram with a time to live of 255. This datagram's data will also be an Active Open to a non-listening port on the IUT. All other datagram values will be valid default values.

- Verification: The IUT should send a reset for each datagram.

- Success: IUT sends a reset for each datagram.

- Failure: The IUT does not return a reset for each datagram.

TEST 110: RECOGNITION OF VERSION NUMBER

Determine that the IUT drops a datagram whose header has an incorrect version number.

- Action: CD will send a datagram with a version number of 3 to the IUT. The datagram's data will be an Active Open to a non-listening port on the IUT. All other datagram values will be valid default values.

- Verification: No response should be received from the IUT and the IUT accepts a subsequent valid datagram sent by the CD.

- Success: No response is received from the IUT and the IUT accepts the subsequent valid datagram sent by the CD.

- Failure: The IUT returns a reset.

TEST 111: RECOGNITION OF INVALID CHECKSUM

Determine that the IUT drops a datagram with an invalid checksum.

- Action: CD will send a datagram with an invalid checksum to the IUT. The datagram's data will be an Active Open to a non-listening port on the IUT. All other datagram values will be valid default values.

- Verification: No response should be received from the IUT and the IUT accepts a subsequent valid datagram sent by the CD.

- Success: No response is received from the IUT and the IUT accepts the subsequent valid datagram sent by the CD.

- Failure: The IUT returns a reset.

TEST 112: RECOGNITION OF ILLEGALLY SMALL HEADER LENGTH

Determine that the IUT drops a datagram with an invalid header length.

- Action: CD will send a datagram with a header length of 4 to the IUT. The datagram's data will be an Active Open to a non-listening port on the IUT. All other datagram values will be valid default values.

- Verification: No response should be received from the IUT and the IUT accepts a subsequent valid datagram sent by the CD.

- Success: No response is received from the IUT and the IUT accepts the subsequent valid datagram sent by the CD.

- Failure: The IUT returns a reset.

TEST 113: RECOGNITION OF INCONSISTENT HEADER LENGTH VS. DATAGRAM TOTAL LENGTH

Determine that the IUT drops a datagram with a header length greater than the total length of the datagram.

- Action: CD will send the IUT a datagram with a header length greater than the total length. The datagram's data will be an Active Open to a non-listening port on the IUT. All other datagram values will be valid default values.

- Verification: No response should be received from the IUT and the IUT accepts a subsequent valid datagram sent by the CD.

- Success: No response is received from the IUT and the IUT accepts the subsequent valid datagram sent by the CD.

- Failure: The IUT returns a reset.

TEST 114: RECOGNITION OF ILLEGALLY SMALL TOTAL LENGTH

Determine that the IUT drops a datagram with an illegally small total length.

- Action: CD will send the IUT a datagram with a total length set smaller than the minimum header length. The datagram's data will be an Active Open to a non-listening port on the IUT. All other datagram values will be valid default values.

- Verification: No response should be received from the IUT and the IUT accepts a subsequent valid datagram sent by the CD.

- Success: No response is received from the IUT and the IUT accepts the subsequent valid datagram sent by the CD.

- Failure: The IUT returns a reset.

TEST 115: RECOGNITION OF INCORRECTLY LARGE TOTAL LENGTH

Determine that the IUT recognizes and drops a datagram with a total length greater than the actual datagram length.

- Action: CD will send the IUT a datagram with a total length set greater than its actual length. The datagram's data will be an Active Open to a non-listening port on the IUT. All other datagram values will be valid default values.

- Verification: No response should be received from the IUT and the IUT accepts a subsequent valid datagram sent by the CD.

- Success: No response is received from the IUT and the IUT accepts the subsequent valid datagram sent by the CD.

- Failure: The IUT returns a reset.

TEST 116: RECOGNITION OF INCORRECTLY SMALL TOTAL LENGTH

Determine that the IUT recognizes and drops a datagram with a specified total length smaller than the actual datagram length.

- Action: CD will send the IUT a datagram with a total length set smaller than its actual length. The datagram's data will be an Active Open to a non-listening port on the IUT. All other datagram values will be valid default values.

- Verification: No response should be received from the IUT and the IUT accepts a subsequent valid datagram sent by the CD.

- Success: No response is received from the IUT and the IUT accepts the subsequent valid datagram sent by the CD.

- Failure: The IUT returns a reset.

TEST 117: RECOGNITION OF MORE FRAGMENTS FIELD

Determine that the IUT recognizes the IP header more fragments field and does not accept a datagram with that field when no other datagram with the same id follows.

- Action: CD will send the IUT a datagram with the more fragments field set. The datagram's data will be an Active Open to a non-listening port on the IUT. All other datagram values will be valid default values.

- Verification: No response should be received from the IUT and the IUT accepts a subsequent valid datagram sent by the CD.

- Success: No response is received from the IUT and the IUT accepts the subsequent valid datagram sent by the CD.

- Failure: The IUT returns a reset.

TEST 118: ACCEPTANCE OF DATAGRAM WITH ONE OPTION

Determine that the IUT accepts a datagram set with an option.

- Action: CD will send the IUT a datagram with one NOP option. The datagram's data will be an Active Open to a non-listening port on the IUT. All datagram values will be valid default values.

- Verification: The IUT should send a reset for the datagram.

- Success: IUT sends a reset for the datagram.
- Failure: The IUT does not return a reset for the datagram.

TEST 119: ACCEPTANCE OF DATAGRAM WITH TWO OPTIONS

Determine that the IUT accepts a datagram set with two NOP options.

- Action: CD will send a datagram with two NOP options to the IUT. The datagram's data will be an Active Open to a non-listening port on the IUT. All datagram values will be valid default values.

- Verification: The IUT should send a reset for the datagram.

- Success: IUT sends a reset for the datagram.

- Failure: The IUT does not return a reset for the datagram.

TEST 120: ACCEPTANCE OF DATAGRAM WITH THREE OPTIONS

Determine that the IUT accepts datagram set with three NOP options.

- Action: CD will send a datagram with three NOP options to the IUT. The datagram's data will be an Active Open to a non-listening port on the IUT. All datagram values will be valid default values.

- Verification: The IUT should send a reset for the datagram.

- Success: IUT sends a reset for the datagram.

- Failure: The IUT does not return a reset for the datagram.

TEST 121: ACCEPTANCE OF DATAGRAM WITH FOUR OPTIONS

Determine that the IUT accepts a datagram set with four NOP options.

- Action: CD will send a datagram with four NOP options to the IUT. The datagram's data will be an Active Open to a non-listening port on the IUT. All datagram values will be valid default values.

- Verification: The IUT should send a reset for the datagram.

- Success: IUT sends a reset for the datagram.

- Failure: The IUT does not return a reset for the datagram.

TEST 122: RECOGNITION OF INVALID OPTION TYPES

Determine that IUT drops a datagram with invalid option types.

- Action: Central Driver will send the IUT a datagram with four undefined options. The datagram's data will be an Active Open to a non-listening port on the IUT. All datagram values will be valid default values.

- Verification: The IUT should send no response to the datagram.

- Success: IUT sends no response to the datagram and accepts a subsequent valid datagram.

- Failure: The IUT returns a reset for the datagram.

===

FRAGMENTS Scenario

This scenario will test that the IUT can reassemble fragments of the same datagram and that the IUT checks the fields of the fragment datagrams for validity.

TEST 123: REASSEMBLY OF TWO-FRAGMENT DATAGRAM

Determine that the IUT can reassemble a datagram sent in two fragments.

- Action: CD will send the IUT a datagram divided into two fragments. The datagram's data will be an Active Open to a non-listening port on the IUT. All datagram values will be valid default values.

- Verification: The IUT should send a reset for the datagram.

- Success: IUT sends a reset for the datagram.

- Failure: The IUT does not return a reset for the datagram.

TEST 124: REASSEMBLY OF THREE-FRAGMENT DATAGRAM

Determine that the IUT can reassemble a datagram sent in three fragments.

- Action: CD will send a datagram divided into three fragments to the IUT. The datagram's data will be an Active Open to a non-listening port on the IUT. All datagram values will be valid default values.

- Verification: The IUT should send a reset for the datagram.

- Success: IUT sends a reset for the datagram.

- Failure: The IUT does not return a reset for the datagram.

TEST 125: REASSEMBLY OF 576-OCTET DATAGRAM

Determine that the IUT can reassemble a datagram totaling 576 octets.

- Action: CD will send the IUT a 576-octet datagram divided into three fragments. The datagram's data will be an Active Open with Data to a non-listening port on the IUT. All datagram values will be valid default values.

- Verification: The IUT should send a reset for the datagram.

- Success: IUT sends a reset for the datagram.

- Failure: The IUT does not return a reset for the datagram.

TEST 126: REASSEMBLY OF OUT-OF-ORDER FRAGMENTS -- MIXED

Determine that the IUT can reassemble a datagram whose fragments are sent out of order.

- Action: CD will send a datagram divided into four fragments to the IUT. The fragments will be sent out of order. The datagram's data will be an Active Open with Data to a non-listening port on the IUT. All datagram values will be valid default values.

- Verification: The IUT should send a reset for the datagram.

- Success: IUT sends a reset for the datagram.

- Failure: The IUT does not return a reset for the datagram.

TEST 127: REASSEMBLY OF OUT-OF-ORDER FRAGMENTS -- REVERSED

Determine that the IUT can reassemble a datagram whose fragments are sent out of order.

- Action: CD will send a datagram divided into four fragments to the IUT. The fragments will be sent in reverse order (the final fragment first). The datagram's data will be an Active Open with Data to a non-listening port on the IUT. All datagram values will be valid default values.

- Verification: The IUT should send a reset for the datagram.

- Success: IUT sends a reset for the datagram.

- Failure: The IUT does not return a reset for the datagram.

TEST 128: RECOGNITION OF ARRIVING FRAGMENT'S EXPIRED TIME TO LIVE

Determine that the IUT does not reassemble a datagram with a fragment whose time to live has expired on arrival.

- Action: CD will send a datagram divided into fragments to the IUT. One fragment will have an expired time to live. The datagram's data will be an Active Open with Data to a non-listening port on the IUT. All datagram values will be valid default values.

- Verification: No response should be received from the IUT and the IUT should accept a subsequent valid datagram sent by the CD.

- Success: No response is received from the IUT and the IUT accepts the subsequent valid datagram sent by the CD.

- Failure: The IUT returns a reset.

TEST 129: REASSEMBLY WITH DUPLICATE FRAGMENTS

Determine that the IUT can reassemble a datagram when a fragment is duplicated.

- Action: CD will send a datagram divided into three fragments to the IUT. One fragment will be sent twice. The datagram's data will be an Active Open with Data to a non-listening port on the IUT. All datagram values will be valid default values.

- Verification: The IUT should send a reset for the datagram.

- Success: The IUT sends a reset for the datagram.

- Failure: The IUT does not return a reset for the datagram.

TEST 130: CHECKING PROTOCOL FIELDS IN FRAGMENTS FOR CONSISTENCY

Determine that the IUT does not reassemble a datagram when the protocol fields of the fragments are not the same.

- Action: CD will send a datagram divided into fragments to the IUT. One fragment will have a protocol number other than the TCP protocol number. The datagram's data will be an Active Open with Data to a non-listening port on the IUT. All datagram values will be valid default values.

- Verification: No response should be received from the IUT and the IUT should accept a subsequent valid datagram sent by the CD.

- Success: No response is received from the IUT and the IUT accepts the subsequent valid datagram sent by the CD.

- Failure: The IUT returns a reset.

TEST 131: CHECKING OF FRAGMENT PRECEDENCE FIELDS FOR CONSISTENCY

Determine that the IUT does not reassemble a datagram when the precedence fields of the fragments are not the same.

- Action: CD will send a datagram divided into fragments to the IUT. One fragment will have a different precedence. The datagram's data will be an Active Open with Data to a non-listening port on the IUT. All datagram values will be valid default values.

- Verification: No response should be received from the IUT and the IUT should accept a subsequent valid datagram sent by the CD.

- Success: No response is received from the IUT and the IUT accepts the subsequent valid datagram sent by the CD.

- Failure: The IUT returns a reset.

==

EXTENDIP Scenario

The EXTENDIP scenario tests IP functions that require a Remote Driver for verification. This scenario should be run after BASIC, the initial TCP scenario that will test simple TCP OPEN, CLOSE, and data transfer. EXTENDIP tests the IUT's ability to set and restart its reassembly timer; reassemble multiple datagrams; fragment correctly; and set the complete range of precedence values.

TEST 132: EXPIRATION OF FRAGMENT'S TIME TO LIVE DURING REASSEMBLY

Determine that the IUT does not reassemble a datagram when a fragment's time to live expires during reassembly.

- Action: CD will send the IUT a datagram divided into fragments. One fragment will be delayed; one transmitted fragment will have a short time to live of x seconds. The final fragment will be sent after x seconds. The datagram's data will be an Active Open with Data to a non-listening port on the IUT. All datagram values will be valid default values.

- Verification: No response should be received from the IUT and the IUT should accept a subsequent valid datagram sent by the CD.

- Success: No response is received from the IUT and the IUT accepts the subsequent valid datagram sent by the CD.

- Failure: The IUT returns a reset.

TEST 133: SET AND RESTART THE REASSEMBLY TIMER

Determine that the IUT sets and restarts the reassembly timer as the time to live values of the incoming datagram fragments dictate.

- Action: The CD determines that the IUT checks the time to live in a fragment. After a TCP connection has been established, the CD will send a datagram divided into fragments. One fragment will be delayed. One of the first fragments sent will have a time to live of x. A following fragment will have a time of x+y. The delayed fragment will not be sent until x seconds have passed. All datagram values will be valid.

- Verification: The IUT should deliver the data carried by the fragmented datagram.

- Success: The IUT delivers the datagram's data showing that reassembly had taken place.

- Failure: The IUT does not deliver the datagram's data. The test will also fail if the IUT ignores the fragment's time to live field.

TEST 134: REASSEMBLY OF TWO INTERMIXED DATAGRAMS

Determine that the IUT can correctly reassemble more than one datagram concurrently.

- Action: The CD will send two fragmented datagrams to the IUT over an established TCP connection. Fragments from the two datagrams will be intermixed.

- Verification: The IUT should deliver the data from the two datagrams.

- Success: The IUT delivers the data from the two datagrams.

- Failure: The IUT does not deliver the datagram's data.

TEST 135: REASSEMBLY OF MULTI-INTERMIXED DATAGRAMS

Determine that the IUT can correctly reassemble many datagrams concurrently.

- Action: The CD will send four fragmented datagrams to the IUT over an established TCP connection. Fragments from the two datagrams will be intermixed.

- Verification: The IUT should deliver the data from the four datagrams.

- Success: The IUT delivers the data from the four datagrams.

- Failure: The IUT does not deliver the datagram's data.

TEST 136: TRANSMISSION OF ALL VALID PRECEDENCE VALUES

Determine whether the IUT can set the range of precedence values (0-7).

- Action: The CD will instruct the IUT to do a series of Passive Opens with precedence. The precedence will be varied from 0-7. The CD will then instruct the Reference TCP to open a connection to the IUT with the same precedence.

- Verification: After each connection attempt, the IUT SYN ACK will be checked to see if the precedence was set correctly.

- Success: The IUT transmitted all precedence values correctly.

- Failure: The IUT did not transmit all precedence values correctly.

- Inconclusive: The IUT did not allow precedence to be set in its TCP interface call.

===

Scenario ICMP

Scenario ICMP tests the ability of the IUT to accept basic ICMP messages and to generate ICMP replies to these messages.

TEST 137: ECHO REQUEST AND ECHO REPLY ICMP MESSAGES

Determine if the IUT can accept an Echo Request ICMP message and generate an Echo Reply message.

- Action: The CD requests that an Echo Request Message be sent to the IUT.

- Verification: An Echo Reply message should be received from the IUT.

- Success: An Echo Reply message is received from the IUT.

- Failure: An Echo Reply message is not received from the IUT.

TEST 138: TIMESTAMP AND TIMESTAMP REPLY ICMP MESSAGES

Determine if the IUT can accept a Timestamp ICMP message and generate a Timestamp Reply message.

- Action: The CD requests that a Timestamp Message be sent to the IUT.

- Verification: A Timestamp Reply message should be received from the IUT.

- Success: A Timestamp Reply message is received from the IUT.

- Failure: A Timestamp Reply message is not received from the IUT.

TEST 139: INFORMATION REQUEST AND INFORMATION REPLY ICMP MESSAGES

Determine if the IUT can accept an Information Request ICMP message and generate an Information Reply message.

- Action: The CD requests that an Information Request Message be sent to the IUT.

- Verification: An Information Reply message should be received from the IUT.

- Success: An Information Reply message is received from the IUT.

- Failure: An Information Reply message is not received from the IUT.

===

4B - TCP SCENARIOS AND TEST DESCRIPTIONS

Scenario BASIC

Scenario BASIC is the first TCP test of a test session. This scenario tests the most basic TCP functions of the TCP Implementation Under Test (IUT) and the compliance of the IUT Remote Driver (RD) to the TCP Remote Driver Specification. If the IUT and the RD do not receive good results on the first run of this scenario, further testing should be abandoned until the problems exhibited are corrected.

BASIC tests the TCP implementation for its ability to perform the most basic TCP functions: Active Open, Passive Open, transfer of data, and closing. The scenario determines that the Remote Driver interprets Central Driver commands correctly, acknowledges the receipt of a command from the CD, and correctly formats IUT responses that it sends to the CD.

TEST 1: UNSPECIFIED PASSIVE OPEN

Does the IUT implement a Passive Open that accepts an Active Open request?

- Action: RD performs a Specified Passive Open. LSD does Laboratory Slave Driver (LSD) does an Active Open to it.

- Verification: Determine that a connection is made by finding an **OPEN SUCCESS** response from the LSD. Ensure that the RD acknowledges all commands received from the CD. Determine that all RD responses are correctly formatted.

- Success: The connection is made. The RD correctly interprets CD commands, acknowledges CD commands, and formats IUT responses.

- Failure: Connection is not made or the RD does not correctly interpret CD commands, acknowledge CD commands, or format IUT responses.

TEST 2: ACTIVE OPEN

Can the IUT implement an Active Open?

- Action: LSD performs a Passive Open. The RD does an Active Open to the listening port.
- Verification: Determine that a connection is made by finding an **OPEN SUCCESS** response from the RD. Ensure that the RD acknowledges all commands received from the CD. Determine that all RD responses are correctly formatted.
- Success: The connection is made. The RD correctly interprets CD commands, acknowledges CD commands, and formats IUT responses.
- Failure: Connection is not made or the RD does not correctly interpret CD commands, acknowledge CD commands, or format IUT responses.

TEST 3: BASIC DATA TRANSFER

Does the IUT send and deliver data correctly?

- Action: The LSD sends data to the RD. The RD sends data to the LSD.
- Verification: Ensure that all the data that the IUT is directed to send to the LSD is received by the Laboratory Reference Implementation (REF). Check that all the data sent by the LSD is delivered to the RD by the IUT. Ensure that the RD acknowledges all commands received from the CD. Determine that all RD responses are correctly formatted.
- Success: The IUT sends all the data it is requested to send and delivers all the data it receives. The RD correctly interprets CD commands, acknowledges CD commands, and formats IUT responses.
- Failure: The IUT does not send all the data it is requested to send or does not deliver all the data it receives. Also, the RD does not correctly interpret CD commands, acknowledge CD commands, or format IUT responses.

TEST 4: SIGNIFICANCE OF COMMAND LCN TO REMOTE DRIVER AND IUT

Do the IUT and the IUT Remote Driver correctly interpret the local connection name (lcn) specified in the commands the RD receives from the Central Driver?

- Action: The CD sends a Send command to the RD that contains an lcn for a connection that is not established. The RD requests the IUT to send data to the LSD over this connection.
- Verification: Ensure that the RD reports in correct format that the IUT responds to the incorrect lcn with **TCP ERROR: CONNECTION DOES NOT EXIST.**
- Success: The IUT recognizes an invalid lcn. The RD correctly interprets CD commands, acknowledges CD commands, and formats IUT responses.
- Failure: The IUT does not recognize an invalid lcn. Also, the RD does not correctly interpret CD commands, acknowledge CD commands, or format IUT responses.

TEST 5: DETERMINE IUT STANDARD SEND BUFFER FOR TESTING PURPOSES

Determine the standard size buffer that the IUT requires to guarantee that the IUT will output at least two segments without delay.

- Action: The RD issues a series of Send commands; each command requests a larger increment of data to be sent. After each Send command, the LSD waits for the data to be delivered. Once the data is delivered, check the TCP segments collected by the REF to determine if more than one segment was required to send the data. When more than one segment is used to send the data or the response **TCP FRROR: INSUFFICIENT RESOURCES** is returned by the IUT, the test is ended.

- Verification: When a Send command data length requires the IUT to send the data in more than one segment, that data length is noted. If the response **TCP ERROR: INSUFFICIENT RESOURCES** has been reported, the data length of the Send command immediately preceding the command that caused that response is noted.

- Observation: The IUT standard send buffer is noted for the test operator.

TEST 6: CLOSING HANDSHAKE WHEN IUT INITIATES CLOSE

Does the IUT correctly perform the closing handshake when it initiates closing?

- Action: The RD performs a close. When the LSD receives the indication of peer closing from the REF, the LSD performs a close.

- Verification: Ensure that the RD reports **TERMINATE: CONNECTION CLOSED** or **TERMINATE: ULP CLOSE** when the IUT indicates the connection closed. Check the TCP segments collected by the REF to determine that the IUT acknowledged the FIN from the REF before it terminated.

- Success: The IUT acknowledges the FIN of the REF. The IUT reports its connection closed. The RD correctly interprets Central Driver commands, acknowledges CD commands, and formats IUT responses.

- Failure: The IUT does not acknowledge the FIN of the REF or the IUT does not report its connection closed. Also, the IUT reports its connection closed before the REF closes. The RD does not correctly interpret CD commands, acknowledge CD commands, or format IUT responses.

TEST 7: CLOSING HANDSHAKE WHEN IUT PEER INITIATES CLOSE

Does the IUT correctly perform the closing handshake when its peer initiates closing?

- Action: The LSD performs a close. When the IUT reports receiving the REF's closing, it performs a close.

- Verification: Ensure that the RD correctly reports **CLOSING** and **TERMINATE: CONNECTION CLOSED** when it receives these responses from the IUT for the REF's closing and the final closing of the connection. Check the TCP segments collected from the REF to determine that the IUT did not send its FIN before being instructed to close by the RD. Also check that the IUT sends a FIN to the REF once it has been instructed to close.

- Success: The IUT waits to be instructed to close before it sends a FIN. It sends a FIN when it is instructed to close. The IUT correctly reports its peer's closing and the closing of the connection. The RD correctly interprets Central Driver commands, acknowledges CD commands, and formats IUT responses.

- Failure: The IUT sends a FIN in response to a peer's FIN before it is instructed to close; or the IUT does not send a FIN when it is instructed to close. Failure also occurs when the IUT does not report a peer's closing or the closing of the connection; and if the RD does not correctly interpret CD commands, acknowledge CD commands, or format IUT responses.

TEST 8: ABILITY TO RECONNECT TO COMMAND CHANNEL AFTER CLOSURE

Does the IUT Remote Driver remain listening after the Central Driver closes the command channel?

- Action: The Central Driver instructs the Remote Driver to kill itself. After waiting five seconds to allow the connection to be cleared, the CD attempts to reconnect the command channel to the RD.

- Verification: If the scenario aborts when the CD attempts to reconnect the command channel to the RD, the RD is no longer listening. If the CD is able to reconnect the command channel, a Remote Driver is left listening after the command channel is closed.

- Success: A Remote Driver is left listening when the Central Driver closes the command channel to the Remote Driver, and reconnection to the Central Driver occurs.

- Failure: A Remote Driver is not left listening when the Central Driver closes the command channel to the Remote Driver. The RD does not correctly interpret CD commands, acknowledge CD commands, or format IUT responses.

===

Scenario OPEN

Scenario OPEN tests the TCP implementation of Fully Specified Passive Open, and Active Open with Data.

TEST 9: FULLY SPECIFIED PASSIVE OPEN

Does the IUT implement a Passive Open that accepts an Active Open request only from a specified address?

- Action: RD performs a Fully Specified Passive Open. LSD does an Active Open to it from a socket bound with the specified address.

- Verification: Determine that a connection was made by finding an **OPEN SUCCESS** response from the LSD.

- Success: The connection was made.

- Failure: Connection was not made.

TEST 10: FULLY SPECIFIED PASSIVE OPEN

Will the IUT Fully Specified Passive Open accept an Active Open request from an unspecified port?

- Action: RD performs a Fully Specified Passive Open. LSD does an Active Open to it from a socket with a port number different from the one specified in the Fully Specified Passive Open.

- Verification: Determine that a connection was made by checking that the REF sent an OPEN FAILURE response.

- Success: Connection was not made.
- Failure: Connection was made.

TEST 11: ACTIVE OPEN WITH DATA

Does the IUT send data on its SYN segment on an Active Open with Data?

- Action: RD performs an Active Open with Data.
- Verification: Check the IUT's SYN segment to see if data was sent on that segment.
- Success: The IUT sends data on its SYN segment.
- Failure: The IUT does not send data on its SYN segment.

TEST 12: ACTIVE OPEN WITH DATA

Does the IUT acknowledge data received on the SYN segment in its SYN ACK (before connection establishment)?

- Action: LSD performs an Active Open with Data.
- Verification: Check the IUT SYN ACK segment to ensure that it does not acknowledge data.
- Success: Data is not acknowledged on the IUT SYN ACK segment.
- Failure: Data is acknowledged on the IUT SYN ACK segment.

TEST 13: PORT NUMBER RANGE

Can the IUT assign a range of local port numbers?

- Action: RD does two Passive Open requests: one specifies source port 1 and the other specifies source port 65535. The RD also does an Active Open to destination port 65534.

- Verification: Determine whether the IUT returns **SYSTEM ERROR: REQUESTED SOURCE PORT NOT PERMITTED** or the **OPEN ID** from the Passive Open requests. Ensure that a connection was made when destination port 65534 was specified in the Active Open.

- Observation: Result is to indicate the range of port numbers that the IUT allows to be assigned.

TEST 21: MISMATCHED PRECEDENCE

Does the IUT provide the option to set a precedence value for the connection and make correct checks of the precedence of incoming segments to the connection?

- Action: RD passively opens a connection with maximum precedence. LSD actively opens at a less than maximum precedence and REF does not raise its precedence.

- Verification: Determine that IUT sets the precedence option by checking the segments output by the IUT. Ensure that IUT aborts the connection when REF does not match precedence by looking for **TERMINATE: REMOTE ABORT** from the LSD or an **OPEN FAILURE** from the LSD connection.

- Success: IUT sets precedence option and aborts connection when peer's precedence does not match.

- Failure: IUT does not set precedence option or makes connection when precedence does not match.

TEST 22: PRECEDENCE

Does the IUT provide the option to set precedence for the connection and make correct checks on the incoming segments to the connection?

- Action: RD passively opens a connection with maximum precedence. LSD actively opens at a less than maximum precedence. The REF raises its precedence during the opening handshake.

- Verification: Determine that the connection is made by checking that the LSD reports an **OPEN SUCCESS.** Check the IUT segments to ensure that precedence was set.

- Success: The IUT makes the connection and sets precedence.

- Failure: The IUT does not set precedence or does not make the connection, even though the REF matched its precedence.

TEST 23: PRECEDENCE NEGOTIATION

Does the IUT provide the option to set precedence for the connection and raise its own precedence to match its peer's precedence?

- Action: LSD passively opens a connection with a given precedence. The RD opens with a zero precedence.

- Verification: Determine that the connection is made by checking that the LSD reports an **OPEN SUCCESS.** Check the IUT segments to ensure that precedence was raised.

- Success: The IUT makes the connection and sets precedence.

- Failure: The IUT does not raise precedence to match the precedence of the REF.

TEST 24: STATUS

Does the IUT in its status response correctly report a connection's s urce port, destination port, destination address, window, precedence, and state? (These fields were selected to represent both static and dynamic internal information.)

- Action: The RD asks the IUT to give the status of a connection.

- Verification: Ensure that the status response given matches the true status of the connection.

- Success: The IUT reports the status of the connection correctly.

- Failure: The IUT does not report the status of the connection correctly.

Scenario CLOSE

Scenario CLOSE tests the TCP implementation of graceful closings, its handling of aborts, precedence, and status.

TEST 14: DATA TRANSFER IN CLOSING STATES

Does the IUT provide graceful closing by completing data transfer after its Upper Level Protocol has initiated closing?

- Action: The RD sends a large amount of data and immediately closes.
- Verification: Determine that the REF receives all the data the IUT was requested to send.
- Success: All expected data was received by the REF.
- Failure: All expected data not received by REF.

TEST 15: DATA TRANSFER IN CLOSING STATES

Does the IUT provide graceful closing by continuing to send data after receiving peer's FIN segment?

- Action: The IUT is instructed to send data after it has received its peer's closing FIN segment.
- Verification: Determine that the IUT sends the data by checking that all expected data is received by the REF.
- Success: REF receives all data IUT was asked to send.
- Failure: REF does not receive all data IUT was asked to send.

TEST 16: DATA TRANSFER IN CLOSING STATES

Does the IUT deliver data sent after it has initiated closing?

- Action: RD tells IUT to close. On receiving the IUT's close, the LSD requests the REF to send data to the IUT.
- Verification: Determine that the IUT delivers all the data sent by the REF.
- Success: IUT delivers all data sent by REF.
- Failure: IUT does not deliver all data sent by REF.

TEST 17: UPPER LAYER PROTOCOL ABORT

Does the IUT perform an Upper Layer Protocol (ULP) abort by sending a correct reset segment?

- Action: RD aborts the connection.
- Verification: Check that the LSD reports a **TERMINATE: REMOTE ABORT** response and the RD reports **TERMINATE: ULP ABORT** or **TERMINATE: USER ABORT.**
- Success: The IUT sends a reset segment and the correct terminate response.
- Failure: The IUT does not send a reset segment or correct terminate response.

TEST 18: PEER ABORT

Can the IUT detect its peer's abort?

- Action: The LSD instructs its REF to abort connection.
- Verification: Check that the IUT reports **TERMINATE: REMOTE ABORT.**
- Success: IUT correctly reports **REMOTE ABORT**.
- Failure: IUT does not correctly report its peer's abort.

TEST 19: UPPER LAYER PROCESS ABORT

Does the IUT discard data queued for sending when it performs a ULP abort?

- Action: RD sends data to its peer and then immediately aborts the connection.
- Verification: Check that the LSD reports a **TERMINATE: REMOTE ABORT** response and the RD reports **TERMINATE: ULP ABORT**. Examine the IUT reset segment for correctness and determine if all data sent by the RD was received by the LSD.
- Success: The IUT sends the correct terminate response, its reset segment is correctly formatted, and not all data sent by the IUT is received by the REF.
- Failure: The IUT sends an incorrect terminate response or an incorrect reset segment.
- Inconclusive: If the LSD receives all data, this test is inconclusive because the IUT may already have sent the data before receiving the abort command.

TEST 20: PEER ABORT

Does the IUT detect its peer's abort and then discard data queued for sending?

- Action: The LSD instructs its REF to abort while the IUT is sending data.

- Verification: Check that the IUT reports **TERMINATE: REMOTE ABORT** and that the IUT stops sending data after receiving the peer's abort.

- Success: IUT correctly reports **REMOTE ABORT** and stops sending data after receiving its peer's reset.

- Failure: IUT does not correctly report its peer's abort.

- Inconclusive: If the LSD receives all the data sent, this test is inconclusive because the IUT may already have sent the data before receiving the abort Command.

==

Scenario RELIABILITY

Scenario RELIABILITY tests the TCP implementation's ability to maintain data integrity: out-of-order data, overlapping data, lost data, segments with bad checksums, and segments with sequence number wraparound.

TEST 25: OUT-OF-ORDER DATA

Can the IUT correctly handle segments that arrive out of order, as indicated by their sequence numbers?

- Action: The LSD sends data to the IUT. The REF divides the data into segments and outputs them out of order.

- Verification: Determine if the IUT delivers the data in the correct order.

- Success: The IUT correctly reorders the data.

- Failure: The IUT does not correctly reorder the data.

TEST 26: OVERLAPPING DATA

Can the IUT clean up overlapping data?

- Action: The LSD sends data to the IUT. The REF repackages the data for retransmission so that some segments contain both new data and data already sent.

- Verification: Determine that IUT accepts the data and correctly delivers it.

- Success: The IUT is able to accept the overlapping data and deliver it correctly.

- Failure: The IUT does not deliver the data on segments after the first arrival or delivers the data incorrectly.

TEST 27: LOST DATA

Can the IUT detect that data is lost?

- Action: The LSD sends data to the RD. The REF divides the data into several segments and sends data, omitting a segment. The missing segment is not sent until the last segment sent is retransmitted several times.

- Verification: Determine that the IUT does not acknowledge anything sent after this missing segment until the missing segment is transmitted.

- Success: The IUT does not acknowledge any data received after the last correctly ordered segment, until the missing segment is sent.

- Failure: The IUT acknowledges data sent after the missing segment before the missing segment is sent.

TEST 28: CHECKSUM

Does the IUT detect a segment with a bad checksum and discard it?

- Action: The LSD sends data to the IUT. The REF sends out a segment with a bad checksum and retransmits it incorrectly several times before transmitting it correctly.

- Verification: Determine that the IUT acknowledges only the segments sent with a good checksum by checking segment data collected by the REF.

- Success: The IUT acknowledges only segments sent with good checksums.

- Failure: The REF acknowledges segments sent with bad checksums.

TEST 29: SEQUENCE NUMBER WRAPAROUND

Does the IUT use correct module arithmetic when comparing sequence numbers?

- Action: The LSD sends data to the IUT. The REF uses as its initial sequence number a number guaranteed to cause wraparound from 2**32-1 to 0 on the data segments.

- Verification: Determine if the IUT acknowledges all data sent by the REF.

- Success: The IUT acknowledges the data sent by the REF.

- Failure: The IUT does not acknowledge any data sent by the REF after the sequence number wraparound occurs.

Scenario MULTIPLEX

Scenario BASIC 4 tests how well the TCP implementation can establish multiple connections and demultiplex data sent over these connections. A TCP connection is defined by the source port, source address, destination port, and destination address of the connection. This four-element identification is known as a 4-tuple. Scenario MULTIPLEX tests by opening multiple connections with different combinations in the 4-tuple (from the viewpoint of the IUT).

TEST 30: MULTIPLEX DATA OVER 2 CONNECTIONS WITH UNIQUE 4-TUPLES

Can the IUT correctly deliver data sent over 2 connections it actively opened, with no 4-tuple element in common?

- Action: The RD opens 2 connections to the LSD. The LSD then sends unique data over each connection.
- Verification: Determine that the IUT keeps separate the data sent on each connection and delivers it correctly.
- Success: The IUT delivers the data sent over each connection correctly.
- Failure: The IUT does not deliver the data sent over each connection correctly.

TEST 31: MULTIPLEX OVER 2 CONNECTIONS WITH COMMON DESTINATION PORT IN 4-TUPLES

Can the IUT multiplex data over 2 connections that share a common destination port in their 4-tuples?

- Action: Two connections are opened that have the same destination port on the REF. The LSD sends unique data over each connection.
- Verification: Determine that the IUT keeps separate the data received over each connection and delivers it correctly.

- Success: The IUT correctly delivers the data sent over each connection.

- Failure: The IUT does not deliver the data sent over each connection correctly.

TEST 32: MULTIPLEX WITH 2 CONNECTIONS HAVING SAME REF SRC PORT IN 4-TUPLES

Can the IUT multiplex data over 2 connections that have the same destination port in their 4-tuples?

- Action: The LSD actively opens 2 connections from the same source port to distinct listening ports on the IUT. The LSD then sends unique data to the RD over each connection.

- Verification: Determine that all the data sent by the LSD is delivered correctly to the RD.

- Success: All sent data is delivered correctly to the RD.

- Failure: Data is not delivered correctly to the RD.

TEST 33: MULTIPLEX DATA OVER 2 CONNECTIONS WITH UNIQUE 4-TUPLES

Can the IUT correctly deliver data sent over 2 connections that the IUT passively opens, when the connections have no 4-tuple elements in common?

- Action: The LSD opens 2 connections to the RD. The LSD then sends unique data data over each connection.

- Verification: Determine that the IUT keeps separate the data sent on each connection and delivers it correctly.

- Success: The IUT delivers the data sent over each connection correctly.

- Failure: The IUT does not deliver the data sent over each connection correctly.

TEST 34: MULTIPLEX DATA OVER 3 CONNECTIONS WITH SAME SRC PORT IN 4-TUPLE

Can the IUT correctly establish connections and demultiplex data when the same IUT source port is in the 4-tuple of 3 connections?

- Action: The LSD actively opens 3 connections to the same IUT port. The LSD then sends unique data to the RD over each connection.

- Verification: Determine that all 3 connections are opened and that all the data sent over each connection is correctly delivered.

- Success: All the data sent over the 3 connections is correctly delivered.

- Failure: The data sent over the 3 connections is not correctly delivered.

TEST 35: IUT REJECTS DUPLICATE CONNECTION

Does the IUT reject an attempt to *make a duplicate* connection to its listening port?

- Action: The LSD actively opens 2 connections to the IUT from different ports. The LSD then attempts to make a third connection using the same source and destination port used in the second connection.

- Verification: Determine that IUT refuses to make the duplicate connection (**OPEN FAILURE** response received by LSD), and does not enter into the opening handshake for the duplicate connection.

- Success: The IUT rejects the duplicate connection and does not enter into the opening handshake for that connection.

- Failure: The IUT fails to reject the duplicate connection.

TEST 36: MULTIPLEX DATA OVER CONNECTIONS USING THE SAME SEQUENCE NUMBERS

Can the IUT multiplex data when the REF uses the same sequence number on 2 connections ?

- Action: The LSD opens 3 connections to the IUT and sends unique data over the 3 connections to the RD. The REF ensures that the the TCP segments on 2 of the connections have the same sequence number.

- Verification: Determine that all the data sent over each connection is correctly delivered to the RD.

- Success: All the data sent over each connection is correctly delivered.

- Failure: The data sent over each connection is not correctly delivered.

TEST 37: IUT'S REFUSAL TO MAKE DUPLICATE CONNECTION

Does the IUT refuse to make a duplicate connection when it is actively opening?

- Action: The RD makes a connection to the LSD. The RD then attempts to open a second connection with the same 4-tuple (same source and destination ports).

- Verification: Verify that the IUT refuses to make the duplicate connection by responding with **TCP ERROR: CONNECTION ALREADY EXISTS** or **SYSTEM ERROR: REQUESTED SOURCE PORT IN USE.**

- Success: IUT does not make a duplicate connection and does not allocate resources for a duplicate connection.

- Failure: IUT allocates resources for a duplicate connection or does not give proper response to the attempt to open the duplicate connection.

Scenario SECURITY

Scenario SECURITY tests the TCP Implementation Under Test for basic and default security. It tests that the IUT:

- o Allows the security of the connection to be set by the use of parameters in the open service requests;
- o Rejects a connection request with a wrong security level;
- o Aborts the connection if a segment arrives with any mismatch of the security option.

Tests 38 through 46 test basic security. Tests 47 through 50 test default security. Default security checks should be performed by both classified and unclassified hosts. If the IUT does not pass Test 38 or Test 39 (the setting of security in the IUT Active Open and the IUT Passive Open), Tests 40 through 46 are not executed.

TEST 38: IUT'S ABILITY TO SET SECURITY IN ITS ACTIVE OPEN

Is the IUT able to set security in its Active Open?

- Action: The RD opens a connection with the classification and protection parameters supplied for the IUT Active Open. The LSD opens with the same security in its Passive Open.

- Verification: Determine if the connection is opened successfully. If the connection is opened, determine that the security parameters have been set correctly. If the open service request does not return an **OPEN ID**, the response **TCP ERROR: SEC/PREC NOT ALLOWED** or **SYSTEM ERROR: REQUESTED PARAMETER NOT IMPLEMENTED** must be found.

- Success: Connection is successfully opened showing that the security parameters have been correctly set. If the connection is not established, the response **TCP ERROR: SEC/PREC NOT ALLOWED** or **SYSTEM ERROR: REQUESTED PARAMETER NOT IMPLEMENTED** is found.

- Failure: Connection is not established and neither **TCP ERROR: SEC/PREC NOT ALLOWED** nor **SYSTEM ERROR: REQUESTED PARAMETER NOT IMPLEMENTED** is returned from the IUT. Connection is established but the parameters have not been set correctly.

TEST 39: IUT'S ABILITY TO SET SECURITY PARAMETERS IN ITS PASSIVE OPEN

Is the IUT able to set security in its Passive Open ?

- Action: The RD uses the classification and protection parameters that it is given for an IUT Passive Open. The LSD sets the same parameters in its Active Open and attempts to open a connection.

- Verification: Determine that the connection is established. If a connection is opened successfully, then security fields in the TCP segments are checked to make sure that the IUT is setting the security from its input parameter. The REF collects these segments. If a connection is not established, check that the IUT open service response returned the response **TCP ERROR: SEC/PREC NOT ALLOWED** or **SYSTEM ERROR: REQUESTED PARAMETER NOT IMPLEMENTED.**

- Success: The connection is established. Analysis determines that the security parameters were set in the IUT Passive Open. Also, the connection is not established but the IUT returns either **TCP ERROR: SEC/PREC NOT ALLOWED** or **SYSTEM ERROR: REQUESTED PARAMETER NOT IMPLEMENTED** in response to the open service request.

- Failure: Although the connection is established, analysis reveals that the IUT is not setting security but merely matching peer's security. Also, the connection is not established but no proper security-related response is received to the open service request.

TEST 40: IUT'S ABILITY TO SET SECURITY IN ITS SPECIFIED PASSIVE OPEN

Is the IUT able to set security in its Specified Passive Open?

- Action: The RD uses the classifications CONFID and protection DIA as the parameters for an IUT Passive Open. The LSD sets the same parameters in its Active Open *and* attempts to open a connection.

- Verification: Determine that the connection is established. If a connection is opened successfully, the security fields in the TCP segments are checked to make sure that the IUT is setting security from its input parameter. The REF collects these segments. If a connection is not established, check that the IUT open service response returned the response **TCP ERROR: SEC/PREC NOT ALLOWED** or **SYSTEM ERROR: REQUESTED PARAMETER NOT IMPLEMENTED.**

- Success: The connection is established and analysis determines that the security parameters were set in the IUT *Specified Passive Open*. Also, the connection is not established but the IUT returns either **TCP ERROR: SEC/PREC NOT ALLOWED** or **SYSTEM ERROR: REQUESTED PARAMETER NOT IMPLEMENTED** in response to the open service request.

- Failure: Connection is established but analysis reveals that the IUT is not setting security but merely matching peer's security. Also, connection is not established and no proper security-related response is received to the open service request.

TEST 41: SECURE IUT REJECTS CONNECTION TO UNSECURED PEER

Does the secured IUT reject a connection to an unsecured peer?

- Action: The RD with security set in its Active Open opens to the REF. The REF has passively opened without security set.

- Verification: Check that the IUT returns an **OPEN FAILURE** and that the connection is not established.

- Success: The IUT successfully gets an **OPEN FAILURE**.
- Failure: The IUT fails to get an **OPEN FAILURE**.

TEST 42: SECURE IUT REJECTS CONNECTION ATTEMPT OF UNSECURED PEER

Does the secured IUT reject connection attempt of an unsecured host?

- Action: LSD with no security actively opens to the IUT. The IUT has passively opened with security set.
- Verification: Determine that no connection is established by checking for an **OPEN FAILURE** response from the REF.
- Success: The REF gets an **OPEN FAILURE** or the IUT cannot set security parameters.
- Failure: The connection is established.

TEST 43: SECURITY OPTION PLACEMENT IN DATA SEGMENTS

Is the IUT consistent in its security option placement when sending data?

- Action: A secure connection is opened. Both the LSD and the RD send data.
- Verification: Determine that all the data sent by the LSD and the RD is correctly delivered by the other peer. Check the segment information collected by the REF to ensure that the correct security was placed on every segment.
- Success: All data is delivered and the security is correctly placed on each TCP segment.
- Failure: All data is not correctly delivered or the security fields are not correctly placed on every TCP segment.

TEST 44: RESPONSE TO DATA WITH MISMATCHED SECURITY CLASS

Does the IUT reset a connection on receiving data with a mismatched security class?

- Action: A secure connection is established. The LSD sends data to the RD. The REF places a wrong security class on the data segments.

- Verification: Determine that the IUT resets connection by checking that the REF reports **REMOTE ABORT**. Check that the IUT reports **SEC/PREC MISMATCH.**

- Success: The IUT resets the connection and reports **SEC/PREC MISMATCH.**

- Failure: The IUT fails to reset connection or does not report **SEC/PREC MISMATCH.**

TEST 45: RESPONSE TO DATA WITH MISMATCHED PROTECTION AUTHORITY

Does the IUT reset a connection on receiving data with a mismatched security authority?

- Action: A secure connection is established. The LSD sends data to the RD. The REF places an incorrect protection authority on the data segments.

- Verification: Determine that the IUT resets the connection by checking that the REF reports **REMOTE ABORT**. Check that the IUT reports **SEC/PREC MISMATCH.**

- Success: The IUT resets the connection and reports **SEC/PREC MISMATCH.**

- Failure: IUT fails to reset connection or does not report **SEC/PREC MISMATCH.**

TEST 46: RESPONSE TO DATA WITH AN EXTRA PROTECTION AUTHORITY

Does the IUT reset a connection on receiving data with an extra security protection authority?

- Action: A secure connection is established. The LSD sends data to the RD. The REF places an extra protection authority on the data segments.

- Verification: Determine that the IUT resets the connection by checking that the REF reports **REMOTE ABORT.** Check that the IUT reports **SEC/PREC MISMATCH.**

- Success: The IUT resets the connection and reports **SEC/PREC MISMATCH.**

- Failure: IUT fails to reset connection or does not report **SEC/PREC MISMATCH.**

TEST 47: USE OF SECURITY OPTION FOR UNCLASSIFIED CONNECTIONS

Does the IUT use the security option for unclassified connections?

- Action: The LSD does a Passive Open with security set. The RD does an Active Open with no security set.

- Verification: Check TCP segments collected by the REF to determine if IUT uses the security option at a default level for an unsecured connection. Ensure that the connection is not opened by looking for the **OPEN FAILURE** response from the IUT.

- Observation: IUT does or does not use the security option for unclassified connections.

- Failure: Test connection opened.

TEST 48: RECOGNITION OF UNCLASS AND GENSER AS EQUIVALENT TO UNSECURE

Does the unsecured IUT connect with a peer with security of UNCLASS and GENSER?

- Action: LSD does a passive open with security set to UNCLASS and GENSER. The RD does an active open with no security setting.

- Verification: Determine that the connection is opened by looking for **OPEN SUCCESS** response from the IUT.

- Success: Connection established.

- Failure: Connection not established.

TEST 49: UNSECURED IUT RESPONSE TO CONNECTION ATTEMPT BY SECURE HOST

Does the IUT with no security reject an Active Open from a peer with security set?

- Action: RD does a Passive Open with no security set. The LSD does an Active Open with security set to CONFID and DIA.

- Verification: Determine that the IUT rejects the connection by searching for **OPEN FAILURE** response reported by LSD.

- Success: Connection is not established.

- Failure: Connection established.

TEST 50: UNSECURED IUT RESPONSE TO DATA MARKED WITH CLASSIFIED SECURITY

Does the IUT reset connection when finding data marked with security higher than unclassified?

- Action: A connection is established with no security settings. The LSD sends data to the RD. The REF places a classified security option on a data segment.

- Verification: Determine that the IUT resets the connection and reports **SEC/PREC MISMATCH** to the RD.

- Success: The IUT correctly resets connection on finding incorrectly marked security on data.

- Failure: The IUT fails to recognize wrong security marking on data and does not reset the connection.

==

Scenario ALLOC

Scenario ALLOC tests the TCP implementation for its ability to perform the ALLOC service request correctly in the EXPLICIT mode under control of the Central Driver.

TEST 51: ALLOC

Can the IUT perform the ALLOC service request correctly?

- Action: RD and LSD establish a connection. The RD instructs the IUT it has allocated a specified buffer size for data. Then the LSD sends data equal to twice that buffer size to the RD.

- Verification: Determine that the amount of data delivered to the RD by the IUT is not greater than that specified in the ALLOC request. If all the sent data is acknowledged prior to the RD's sending another ALLOC, the amount of data delivered must be equal to or less than the buffer size specified in the ALLOC. If it is necessary for the RD to send a second ALLOC for all the sent data to be acknowledged, then results are determined by examining the TCP segments collected by the REF. The segments are analyzed to make sure that no more data was acknowledged than was specified in the ALLOC before the IUT receive window was set to zero.

- Success: The IUT delivers data equal to or less than the ALLOC request.

- Failure: The IUT delivers more data than ALLOC specifies.

- Inconclusive: Analysis cannot be done to determine IUT's performance.

Scenario POLICY

Scenario POLICY tests the TCP implementation of maximum segment size option, basic retransmission, and ULP timeout.

TEST 52: MAXIMUM SEGMENT SIZE OPTION

Does the IUT use the maximum segment size (MSS) option and respond correctly to its peer's maximum transmission unit?

- Action: The LSD and the RD establish a connection. The REF places a low MSS on its opening segment. The RD sends data.
- Verification: Check segment data collected by the REF to see if IUT uses the MSS option. Examine the segment data to see if the IUT sends any data segment with a length greater than the MSS specified by the REF.
- Success: The IUT sends segments that do not exceed the maximum segment size specified by the REF.
- Failure: IUT sends segments that exceed the maximum segment size specified by the REF.
- Observation: Observation is made as to whether the IUT uses the MSS option.

TEST 53: RETRANSMISSION AFTER ACKNOWLEDGMENT OF DATA

Does the IUT stop retransmitting promptly after receiving acknowledgment of data?

- Action: The LSD and the RD establish a connection. The RD sends data. The LSD withholds acknowledgments until it has received several transmissions of the oldest data.
- Verification: Examine the TCP segments collected by the REF. Count the number of retransmissions of data sent after the data is acknowledged. Ensure that the REF receives only a small number of retransmissions after it has acknowledged data.

- Success: The IUT does not retransmit after data is acknowledged.

- Failure: IUT does not stop retransmitting promptly after data is acknowledged.

TEST 54: RETRANSMISSION AFTER ACKNOWLEDGMENT OF SYN AND FIN

Does the IUT stop retransmitting promptly after its SYN and FIN are acknowledged?

- Action: The LSD actively opens a connection to the RD. The REF withholds acknowledgments of the IUT's SYN segment until it is transmitted several times. Then the RD closes the connection. The REF withholds retransmissions until the IUT's FIN segment is transmitted several times.

- Verification: Examine the TCP segments collected by the REF. Count the number of times the IUT SYN segment is retransmitted after it is acknowledged. Count the number of times the IUT FIN segment is retransmitted after it is acknowledged. Ensure that the REF receives only a small number of retransmissions after it acknowledges SYN or FIN.

- Success: The IUT does not retransmit its SYN and FIN segments after they are acknowledged.

- Failure: The IUT does not stop retransmitting its SYN and FIN segments promptly after they are acknowledged.

TEST 55: IMPLEMENTATION OF ULP TIMEOUT SERVICE IN ACTIVE OPEN

Does the IUT implement ULP timeout when timeout is specified in its Active Open?

- Action: Find the number of retransmissions by the IUT when the IUT's default ULP timeout occurs or when the default timeout of 2 minutes suggested by MIL-STD-1778 is reached. This is done by opening a connection and having the RD send data. The REF does not acknowledge the data. When the connection is aborted (or after 2 minutes), check the TCP segments collected by the REF and count the number of retransmissions for one data element. This becomes the default number of retransmissions. The RD does an Active Open with a small ULP timeout and a timeout action of terminate set. The RD then sends data. The REF withholds acknowledgments. The test continues until the connection terminates or 2 minutes pass.

- Verification: Check the TCP segments collected by the REF. Count the number of times a data element is retransmitted. The number of retransmissions must be significantly smaller than the default number of retransmissions.

- Success: IUT implements ULP timeout if it resets the connection and the number of data retransmissions is significantly smaller than the default number of retransmissions.

- Failure: IUT does not implement ULP timeout. The IUT does not reset the connection or, if the connection is reset, the number of IUT retransmissions indicates that the abort was caused by the TCP default rather than by the prescribed ULP timeout.

TEST 56: IMPLEMENTATION OF ULP TIMEOUT SERVICE IN SEND

Does the IUT implement ULP timeout when timeout is specified in its Send?

- Action: The RD and the LSD establish a connection. The RD sends data with a small ULP timeout and a timeout action of terminate set. The REF withholds acknowledgments. The test continues until the connection terminates or 2 minutes pass.

- Verification: Check the TCP segments collected by the REF. Count the number of times a data element is retransmitted. The number of retransmissions must be significantly smaller than the default number of retransmissions (determined in Test 55).

- Success: IUT implements ULP timeout if it resets the connections and the number of data retransmissions is significantly smaller than the default number of retransmissions.

- Failure: IUT does not implement ULP timeout. The IUT does not reset the connection or, if the connection is reset, the number of IUT retransmissions indicates that the abort was caused by the TCP default rather than by the prescribed ULP timeout.

TEST 57: IMPLEMENTATION OF ULP TIMEOUT SERVICE IN PASSIVE OPEN

Does the IUT implement ULP timeout when timeout is specified in its Passive Open?

- Action: The RD does a Passive Open with a ULP timeout set. The LSD performs an Active Open and a connection is established. The RD then sends data. The REF withholds acknowledgments. The test continues until the connection terminates or 2 minutes pass.

- Verification: Check the TCP segments collected by the REF. Count the number of times a data element is retransmitted. The number of data retransmissions must be significantly smaller than the default number of retransmissions (determined in Test 55).

- Success: IUT implements ULP timeout if it resets the connections and the number of data retransmissions is significantly smaller than the default number of retransmissions.

- Failure: IUT does not implement ULP timeout. The IUT does not reset the connection or, if the connection is reset, the number of IUT retransmissions indicates that the abort was caused by the TCP default rather than by the prescribed ULP timeout.

TEST 58: IMPLEMENTATION OF ULP TIMEOUT NOTIFY SERVICE

Does the IUT implement a ULP timeout where the timeout action is notify?

- Action: The RD does an Active Open with a ULP notify timeout set and establishes a connection with the LSD. The RD sends data. The REF withholds acknowledgments on the data. The test continues until the connection terminates or 2 minutes pass.
- Verification: Check that the RD reports the TCP ERROR message **ULP NOTIFY.**
- Success: IUT implements ULP notify.
- Failure: IUT does not implement ULP notify.

==

Scenario OUT_OF_BAND

Scenario OUT_OF_BAND tests the TCP implementation of zero window, urgent data, and pushed data.

TEST 59: RESPONSE TO ZERO WINDOW

Does the IUT correctly respond to a peer's zero window?

- Action: A connection is established between the LSD and the RD. The RD sends data to the LSD. The REF acknowledges the first data segment but announces a zero window in this acknowledgment.
- Verification: Check the TCP segments collected by the REF for the amount of data sent by the IUT while the REF had a zero window. The IUT should not send more than one byte of data while the REF has a zero window.
- Success: The IUT sends no segment of length greater than 1 while its peer has a zero window.
- Failure: The IUT sends a segment of length greater than 1 while its peer has a zero window.

TEST 60: IMPLEMENTATION OF URGENT SERVICE

Does the IUT set the urgent flag and urgent pointer when requested to send urgent data?

- Action: The LSD and the RD establish a connection. The RD sends urgent data to the LSD.

- Verification: Check the TCP segment information collected by the REF to ensure that the IUT sets the urgent flag on every segment of the data. Also check that the value of the urgent pointer on these segments equals the amount of urgent data that exists between the first sequence number of the segment to the end of the urgent data.

- Success: The IUT correctly sets the urgent flag and urgent pointer on urgent data.

- Failure: The IUT does not correctly set the urgent flag or urgent pointer on urgent data.

TEST 61: URGENT SERVICE WHEN PEER HAS ZERO WINDOW

Does the IUT handle urgent correctly when its peer has a zero window?

- Action: The LSD and the RD establish a connection. The RD sends normal data and then urgent data to the LSD. The REF sets its receive window to zero when it acknowledges the first IUT segment. The REF opens its window again later in the data transfer.

- Verification: Check the TCP segments collected by the REF. Ensure that, while the REF is showing a zero window, the IUT sends segments with the urgent flag set. The value of the urgent pointer is set to the number of bytes of urgent data to be sent, and the length is no greater than 1.

- Success: The IUT sends one-byte probe segments and correctly sets the urgent flag and urgent pointer in them.

- Failure: The IUT does not send one-byte probe segments although urgent data is present, or it does not correctly set the urgent flag or urgent pointer in the probe segments it sends.

TEST 62: DIFFERENTIATION BETWEEN URGENT AND NON-URGENT DATA

Is the IUT able to deliver urgent data?

- Action: The LSD and the RD establish a connection. The LSD sends one byte of urgent data to the RD. It then sends the RD some bytes of non-urgent data.

- Verification: Evaluate the **DELIVER** responses from the RD. The urgent data must be delivered in a separate **DELIVER** from the non-urgent data.

- Success: The RD **DELIVER** responses show the one byte of urgent data delivered separately from the subsequent non-urgent data.

- Failure: The RD **DELIVER** responses do not show the one byte of urgent data delivered separately from the subsequent non-urgent data.

TEST 63: PUSH SERVICE WHEN NOT REQUESTED

Does the IUT push data when not requested to do so?

- Action: The LSD and the RD establish a connection. The RD sends data to the LSD. It does not request that the data be pushed.

- Verification: Examine the TCP segments collected by the REF. The push flag should not be set on any segments sent by the IUT, except possibly the last data segment. (It is permissible for the IUT to set a push flag on the very last data segment sent.)

- Success: The IUT correctly does not push any data or the IUT pushes only the last data segment.

- Failure: The IUT incorrectly sets push flags on data not requested to be pushed.

TEST 64: PUSH SERVICE ON REQUEST

Is the IUT able to push data on request?

- Action: The LSD and the RD establish a connection. The RD sends some bytes of data with a push indication. It then sends data without the push indication.

- Verification: Examine the TCP segments collected by the REF. The first data segment the IUT sends should have the push flag set and should contain at least the number of bytes of data sent with the push indication. The only other segment that could have a valid push flag is the last data segment.

- Success: The IUT correctly sets push flag only on pushed data and possibly the last data segment.

- Failure: The IUT fails to set push flag on pushed data or sets push flag on unpushed data.

- Observation: The IUT allows its Upper Level Protocol (ULP) to request rather than command push service. This observation is made when the segment with the push flag carries more than the pushed data.

==

Scenario RESET

Scenario RESET tests whether the TCP implementation does correct reset processing by testing the common reset conditions.

TEST 65: RESPONSE TO CONNECTION REFUSAL

Does the IUT respond correctly to connection refusal?

- Action: The RD actively opens. There is no listening process at the destination port of its Active Open.

- Verification: Determine that the RD receives an **OPEN FAILURE** response from the IUT.

- Success: The IUT reports an **OPEN FAILURE.**

- Failure: The IUT does not report an **OPEN FAILURE.**

TEST 66: PARTIAL RESET PRIOR TO CONNECTION ESTABLISHMENT

Does the IUT continue listening after receiving a reset during connection establishment?

- Action: The RD initiates a Passive Open. The LSD actively opens. On receipt of the IUT's SYN ACK, the REF resets the connection. The LSD attempts to open the connection again.

- Verification: Determine that the LSD reports an **OPEN SUCCESS** response from the REF on the second connection.

- Success: The second connection is established.

- Failure: The second connection cannot be established.

TEST 67: RESPONSE TO RESET RCVD WHILE SENDING DATA OVER CONNECTION

Does the IUT abort with the correct responses when its peer aborts while the IUT is sending data over a connection?

- Action: The LSD and the RD establish a connection. The RD sends data. The REF resets the connection when it receives the first IUT data segment.

- Verification: Determine that, after the reset, the RD reports the response **TERMINATE: REMOTE ABORT** from the IUT and the LSD reports the response **TERMINATE: SERVICE FAILURE** from the REF.

- Success: The correct **TERMINATE** responses are reported from both the IUT and the REF.

- Failure: The correct **TERMINATE** responses are not received from the IUT and the REF.

TEST 68: RESET FORMAT RESPONDING TO ACTIVE OPEN WITHOUT LISTENING PORT

Does the IUT send a correct reset segment on receipt of a SYN for a non-existent port?

- Action: The LSD initiates an Active Open to a port on the IUT without a listening process.

- Verification: Check the TCP segments collected by the REF. The IUT reset segment must have the format: sequence number = 0, acknowledgment number = the sequence number of the REF's SYN segment +1; also, the ACK and RESET flags must be set.

- Success: The IUT uses correct reset segment format.

- Failure: The IUT uses incorrect reset segment format.

TEST 69: RESET FORMAT RESPONDING TO ACTIVE OPEN WITH DATA WITHOUT LISTENING PORT

Does the IUT send a correct reset when it receives the SYN of an Active Open with Data for a non-existent port?

- Actior: The LSD initiates an Active Open with Data to a port on the IUT without a listening process.

- Verification: Check the TCP segments collected by the REF. The IUT reset segment must have the format: sequence number = 0; acknowledgment number = the sequence number of the last byte of data on the REF's SYN segment; and the ACK and RESET flags are set. If the IUT has not passed the Active Open with Data test, the reset segment must have the expected format of Test 68.

- Success: IUT uses correct reset segment format.

- Failure: IUT uses incorrect reset segment format.

TEST 70: RESET FORMAT ON RECEIPT OF INVALID SEGMENT WITH ACK SET

Does the IUT send the correct reset format on receiving a segment with ACK set for a non-existent port?

- Action: The LSD initiates an Active Open to a port on the IUT without a listening process. The REF omits the SYN from its initial segment but sets the acknowledgment flag and puts a value in the acknowledgment number.

- Verification: Check the TCP segments collected by the REF. The IUT's reset must have the format: sequence number = acknowledgment number on the REF's initial segment; also the RESET flag is set. The ACK flag must not be set.

- Success: The IUT uses correct reset segment format.

- Failure: The IUT uses incorrect reset segment format.

TEST 71: RESET FORMAT ON RECEIPT OF INVALID SEGMENT WITH SYN AND ACK SET

Does the IUT send a correct reset on receiving a SYN segment with ACK set for a port in LISTEN state?

- Action: The RD does a Passive Open. The LSD does an Active Open. The REF places an acknowledgment on its SYN segment.

- Verification: Determine that no connection is established. Check that an **OPEN FAILURE** is reported from the REF. If the connection attempt correctly fails, check the TCF segments collected by the REF. The IUT reset must have the following format: sequence number = acknowledgment number on the REF's initial segment, and the RESET flag is set. The ACK flag must not be set.

- Success: The IUT resets and uses correct reset format.

- Failure: The IUT establishes connection after receiving invalid SYN segment, or the IUT uses incorrect reset segment format.

TEST 72: NO RESET ON RECEIPT OF SEGMENT WITH BAD ACK

Does the IUT erroneously perform a reset on receiving a segment with a bad acknowledgment number?

- Action: The LSD and the RD establish a connection. The LSD sends data to the the RD. The REF places an incorrect acknowledgment number on an outgoing segment. The REF retransmits the bad segment three times and then corrects it.

- Verification: Determine that the IUT does not terminate the connection. If the connection is not terminated, check the TCP segments collected by the REF to determine that the IUT has transmitted empty acknowledgments. These acknowledgments must have the format: sequence number = IUT's current segment sequence number; acknowledgment number = the sequence number of the REF's segment (not incremented to acknowledge any of the data on the bad segment); and the ACK flag must be set.

- Success: The IUT sends empty acknowledgments until it receives the corrected segment. The connection is not reset.

- Failure: The IUT resets the connection on receiving the bad data segment or the IUT acknowledges data segments with bad acknowledgment number.

===

Scenario QUAL

Scenario QUAL tests the TCP implementation to determine how many TCP connections it is able to provide. The scenario tests up to 145 connections.

TEST 73: NUMBER OF TCP CONNECTIONS RESOURCES CAN SUPPORT

Determine the maximum number of TCP connections provided by the IUT.

- Action: The LSD performs 12 Passive Opens. The RD is instructed to consecutively open 12 active connections to each of these ports until it runs out of resources.

- Verification: Count every connection where an **OPEN SUCCESS** is the response received on the IUT Active Open Request. When the response **TCP ERROR: INSUFFICIENT RESOURCES** is found or 144 connections are opened, the test is ended. The total number of connections the IUT can support equals the number of connections the RD has opened plus the connection for the command channel.

- Observation: The total number of connections the IUT is able to support is noted. If the IUT is able to support more than 145 connections (it does not run out of resources), this observation is noted.